Tipos de Bases de Datos NoSQL

1. **Bases de datos de documentos**
    - **Ejemplos:** MongoDB, CouchDB
    - **Ventajas:**
        - **Flexibilidad en el esquema:** Los documentos pueden tener estructuras variadas, lo que permite manejar datos semi-estructurados o no estructurados.
        - **Fácil de escalar horizontalmente:** Se puede distribuir datos en múltiples servidores fácilmente.
        - **Consulta poderosa:** Permiten consultas complejas a través de índices secundarios.
    - **Desventajas:**
        - **Consistencia eventual:** En sistemas distribuidos, puede que no siempre se tenga una consistencia inmediata.
        - **Curva de aprendizaje:** Puede ser complicado para quienes están acostumbrados a bases de datos relacionales.
2. **Bases de datos de clave-valor**
    - **Ejemplos:** Redis, DynamoDB
    - **Ventajas:**
        - **Alta velocidad:** Son extremadamente rápidas para operaciones de lectura y escritura.
        - **Simplicidad:** Su modelo de datos es simple y directo.
        - **Escalabilidad:** Fácil de escalar horizontalmente.
    - **Desventajas:**
        - **Consultas limitadas:** Las consultas suelen estar limitadas a búsquedas simples por clave.
        - **Falta de estructura:** No son adecuadas para datos con relaciones complejas.
3. **Bases de datos de columnas anchas**
    - **Ejemplos:** Cassandra, HBase
    - **Ventajas:**
        - **Optimización para grandes volúmenes de datos:** Muy eficaces para escribir y leer grandes cantidades

de datos distribuidos en múltiples nodos. - **Alta disponibilidad y escalabilidad:** Diseñadas para trabajar en entornos distribuidos, permiten escalabilidad horizontal y tolerancia a fallos. - **Escrituras rápidas:** Son eficientes para realizar escrituras masivas de datos.

- **Desventajas:**
    - **Curva de aprendizaje pronunciada:** Puede ser compleja de configurar y mantener.
    - **Modelado de datos complicado:** Requiere un buen entendimiento del modelo de datos para optimizar el rendimiento.

4. **Bases de datos de grafos**
    - **Ejemplos:** Neo4j, ArangoDB
    - **Ventajas:**

        - **Modelado natural de relaciones:** Excelentes para aplicaciones que requieren navegar y analizar relaciones complejas entre datos, como redes sociales o sistemas de recomendaciones.
        - **Consultas eficientes de grafos:** Permiten realizar consultas complejas de grafos de manera eficiente.
        - **Flexibilidad en el esquema:** Permiten agregar y modificar relaciones sin necesidad de una reestructuración significativa de la base de datos.
    - **Desventajas:**
        - **Escalabilidad limitada:** Algunas bases de datos de grafos pueden tener problemas de escalabilidad horizontal.
        - **Curva de aprendizaje:** Requiere un conocimiento específico sobre teoría de grafos y su modelado.

Ventajas Generales de las Bases de Datos NoSQL

- **Escalabilidad horizontal:** Capacidad de escalar agregando más servidores en lugar de mejorar el hardware de un solo servidor.
- **Flexibilidad del esquema:** Los esquemas pueden evolucionar con facilidad sin necesidad de una reestructuración masiva de la base de datos.
- **Alto rendimiento:** Optimizadas para grandes volúmenes de datos y altas tasas de transacciones.
- **Distribución geográfica:** Soportan la distribución de datos en diferentes ubicaciones geográficas.

Desventajas Generales de las Bases de Datos NoSQL

- **Consistencia eventual:** Muchas bases de datos NoSQL sacrifican la consistencia inmediata para obtener disponibilidad y particionamiento tolerante a fallos (según el teorema CAP).
- **Limitaciones en las consultas:** Algunas bases de datos NoSQL tienen capacidades de consulta más limitadas en comparación con las bases de datos SQL.
- **Curva de aprendizaje:** Migrar de una base de datos relacional a NoSQL puede requerir un cambio significativo en el enfoque y en el modelado de datos.
- **Falta de estandarización:** No hay un estándar único para NoSQL, lo que puede hacer que la portabilidad entre sistemas sea difícil.

TERMINOS:

Proceso Extract Transform Load (ETL):

- Extract: Consiste en obtener los datos con los que vamos a trabajar, bien seas extraidos desde una rchivo, base de datos, directamenete de internet (web scrapping) u otras fuentes. Para ello Power BI proporciona lo que se llaman conectores que permiten migrar datos de una fuente externa a Power BI.

- Transform: Consiste en preparar los datos obtenidos en la fase de extract para poder trabajar con ellos y hacer representaciones útiles dentro de Power BI. Es la parte más importante dentro del proceso ETL y la que mayor conocimiento sobre Microsoft Excel, Microsfot Access y macros de Excel requiere, ya que si no preparamos bien los datos en esta fase se nos presentarán problemas a la hora de re realizar la presentación gráfica de estos. Aunque la sintaxis es simialr a las macros de Excel, Power BI emplea su propio lenguaje de programación, llamado DAX. Power BI también nos proporciona una herramienta propia para realizar la etapa de transformación de datos, llamada Power Query, que además de permitirnos procesar y transformar los datos extraidos, también mantendrá un registro de todas las transformaciones que hemos realizado sobre ellos, de manera que si estos cambian en el origen no será necesario volver a repetir todas las operaciones sobre ellos, se harán de forma automática manteniendo los datos actualizados.

- Load: Una vez obtenidos y preparados los datos para su representación, en esta etapa cargamos los datos a la interfaz gráfica de Power BI para poder trabajar con ellos y realizar todo tipo de representaciones gráficas e interactivas. Cabe destacar que gracias a la herramienta Power Query, si los datos cambian en el archivo/lugar de origen de estos, todos los datos de las representaciones gráficas se actualizaran con los nuevos, es decir, funciona de forma dinámica.

OLTP: Se centra en las transacciones y por tanto en operaciones de escritura

OLAP: Se centra en el analisis y la toma de decisiones estrategicas

ACID: Atomicidad Consistencia Isolated (aislamiento) Durabilidad

Atomicidad (Atomicity): En el modelo ACID, la atomicidad asegura que una transacción se ejecute como una unidad atómica. Esto significa que la transacción ocurrirá en su totalidad o no ocurrirá en absoluto. Si una parte de la transacción falla, la totalidad de la transacción se revierte a su estado inicial. Por ejemplo, en una transferencia de fondos entre dos cuentas bancarias, la atomicidad garantiza que tanto el débito de una cuenta como el crédito en la otra se efectúen como una operación única2.

Consistencia (Consistency): La consistencia asegura que una transacción lleve la base de datos de un estado consistente a otro consistente. En otras palabras, las transacciones deben respetar las reglas de integridad definidas en la base de datos. Si una transacción viola alguna regla, se revierte y no se aplica. Por ejemplo, en una base de datos de un sistema de gestión de bibliotecas, la consistencia garantiza que no se pueda asignar un libro a un estante que no existe o a un género no definido2.

Aislamiento (Isolation): La propiedad de aislamiento asegura que una transacción en ejecución sea invisible para otras transacciones hasta que se complete. Esto evita interferencias entre transacciones concurrentes y garantiza que cada transacción se ejecute de manera independiente. El nivel de aislamiento puede variar según la configuración de la base de datos, pero el principio subyacente es que el resultado de una transacción no afecta al resultado de otras transacciones simultáneas2.

Durabilidad (Durability): Una vez que una transacción ha finalizado, los cambios realizados perdurarán en el tiempo. La durabilidad asegura que los datos modificados se mantengan incluso en caso de fallos del sistema o reinicios. Por ejemplo, si se confirma una reserva de vuelo, esa reserva debe permanecer en la base de datos incluso si ocurre un corte de energía o un fallo del servidor2.

Base de Datos Relacional:

Características:

Los datos se organizan en tablas con filas y columnas.

Utiliza el lenguaje de consulta estructurado (SQL) como interfaz estándar.

Casos de Uso:

Aplicaciones empresariales: Ideal para sistemas de gestión de inventario, recursos humanos, contabilidad y ventas.

Sitios web y aplicaciones móviles: Almacena datos estructurados como perfiles de usuarios, publicaciones y comentarios.

Base de Datos NoSQL (No Relacional):

Características:

No sigue el esquema tabular de las bases de datos relacionales.

Utiliza otros lenguajes de programación para consultas.

Casos de Uso:

Big Data y análisis de datos no estructurados: Adecuado para manejar grandes volúmenes de datos distribuidos.

Aplicaciones web y móviles escalables: Útil para almacenar datos flexibles como redes sociales, registros de eventos y recomendaciones personalizadas.

Base de Datos Centralizada:

Características:

Todos los datos residen en un solo servidor.

Fácil de administrar y mantener.

Casos de Uso:

Pequeñas empresas: Para aplicaciones internas o sitios web con tráfico moderado.

Base de Datos Distribuida:

Características:

Los datos se almacenan en múltiples servidores.

Escalable y tolerante a fallos.

Casos de Uso:

Grandes empresas y aplicaciones globales: Proporciona alta disponibilidad y rendimiento.

Base de Datos Orientada a Objetos:

Características:

Almacena objetos complejos con atributos y métodos.

Se utiliza en sistemas que requieren modelado de objetos.

Casos de Uso:

Sistemas de información geográfica (GIS): Para representar datos espaciales y geográficos.

Aplicaciones científicas y de ingeniería: Donde los objetos tienen propiedades específicas.

Base de Datos Gráfica:

Características:

Utiliza estructuras de grafo para representar relaciones entre datos.

Eficiente para consultas de relaciones complejas.

Casos de Uso:

Redes sociales: Modela conexiones entre usuarios, intereses y amistades.

Recomendaciones personalizadas: Analiza patrones de interacción.

Base de Datos en la Nube:

Características:

Los datos se almacenan en servidores en la nube.

Escalable y accesible desde cualquier ubicación.

Casos de Uso:

Aplicaciones web y móviles escalables: Proporciona flexibilidad y escalabilidad.

Colaboración en línea: Almacena documentos compartidos y datos colaborativos.